Com Sci 221 — Programming Languages
Assignment #3 — Winter 1993
Due Wednesday, 3 February

1. You may do this problem in Pascal, C, Scheme or any other programming language(s) with the appropriate constructs. Given a sequence of nonnegative integers (you may choose the representation of this sequence as an array, linked list, file of characters, etc.) find the largest. Produce two different programs, one based on the iteration

   $biggest := 0$;
   start before the 1st list element;
   While there is more in the list **do**
       advance the list, assigning *nextint* the next integer value in the list;
       $biggest := max\ (nextint, biggest)$
   end while

   and the other based on the recursion

   $maxlist(list) =$
       if $(list = nil)$
           then 0
           else $max(first(list), maxlist(rest(list)))$

   Make the following 3 separate and independent modifications, in each case choosing the more appropriate of the two programs above as your starting point.

   (a) Allow negative integers in the input.

   (b) Make the program interactive, so that whenever the user types "`m`", the biggest integer input so far is output.

   (c) Generalize the program to compute the *minmax* of a tree. Minmax is an important game-theoretic function in which, at subsequent levels of a tree, alternate minima and maxima are taken to compute the effects of two players alternating turns, and each applying his or her best possible strategy.

   Explain your method in each case, including the reasons for your choice of a starting point. As usual, you must devise a sequence of inputs that demonstrate the important qualities of your programs.

2. Use the programming language ML to implement the Fibonacci function in the following 4 different ways.

   (a) Encode the usual recursive definition as directly as possible:

   $$\mathit{fib}(0) = 1$$

   $$\mathit{fib}(1) = 1$$

   $$\mathit{fib}(n+2) = \mathit{fib}(n) + \mathit{fib}(n+1)$$

   See p. 41 of the text.

   (b) Take advantage of the fact that arrays and functions in programming languages both correspond to the abstract idea of function. Create an array called *fib*, and store the first 100 values of the Fibonacci function into it. Then, compute values of the Fibonacci function on demand by simple table look-up.

   (c) Notice that each array entry in the last solution was computed from only the previous two entries in the array, ignoring all earlier ones. Take advantage of this fact to develop an iterative (while-loop) implementation of the Fibonacci function, using two integer variables to store two adjacent values of the function.

   (d) Rewrite the last solution tail-recursively. You should get essentially the program of problem 2.6, p. 59 of the text.

3. Write a short essay (one or two pages should suffice) on the relationship between the solutions above. In particular, focus on their relative advantages and disadvantages in clarity, efficiency, and any other characteristics that seem important. Explain why you are confident that all 4 methods compute the same function. Focus on the fact that the same integer data values are being computed from one another in the same way, but the control of that computation is varying. Your explanation should show exactly where and/or when each value of the Fibonacci function appears in each program.